Synthesis of Tamper-Resistant Pin-Constrained Digital Microfluidic Biochips

Jack Tang, *Member, IEEE*, Mohamed Ibrahim, *Member, IEEE*, Krishnendu Chakrabarty, *Fellow, IEEE*, and Ramesh Karri, *Senior Member, IEEE*

***Abstract*—Digital microfluidic biochips (DMFBs) are an emerging technology that implements bioassays through manipulation of discrete fluid droplets. Recent results have shown that DMFBs are vulnerable to actuation tampering attacks, where a malicious adversary modifies control signals for the purposes of manipulating results or causing denial-of-service. Such attacks leverage the highly programmable nature of DMFBs. However, practical DMFBs often employ a technique called *pin mapping* to reduce control pin count while simultaneously reducing the degrees of freedom available for droplet manipulation. Attempts to control specific electrodes as part of an attack cannot be made without inadvertently actuating other electrodes on-chip, which makes the tampering evident. This paper explores this tamper resistance property of pin mapping in detail. We derive relevant security metrics, evaluate the tamper resistance of several existing pin mapping algorithms, and propose a new security-aware pin mapper. Further, we develop integer linear programming-based methodologies for inserting indicator droplets into a DMFB in order to boost tamper resistance. Experimental results show that the proposed techniques can significantly increase the difficulty for an attacker to make stealthy changes to the execution of a bioassay.**

***Index Terms*—Digital microfluidics, electrode addressing, indicator droplets, integer linear programming (ILP), security, tamper resistance.**

I. Introduction

DIGITAL microfluidic biochips (DMFBs) are platforms for biochemical assays that manipulate fluids in discrete quantities [2]. DMFB technology has made significant strides over the last decade, as their reprogrammable nature is amenable to advanced design automation techniques [3]. Unfortunately, DMFBs are susceptible to actuation tampering attacks—i.e., malicious modifications of control signals—which can achieve disastrous outcomes, such as Denial-of-Service (DoS) and assay result manipulation [4]. With the recent commercialization of DMFB systems, such as the Baebies SEEKER [5], and high-profile incidents, such as the violation of diagnostic integrity at Theranos [6], it is clear that now is a critical time to ensure the security and trustworthiness of microfluidic platforms.

DMFBs typically operate on the principle of electrowetting-on-dielectric [7]. Fluid droplets on a hydrophobic surface change their contact angle when an electric field is applied to the underlying electrode (Fig. 1). By placing a patterned grid of electrodes on a substrate and carefully applying a sequence of control-voltage activations and deactivations, operations, such as dispensing, transporting, mixing, and spitting of droplets can be achieved [8]. These operations can then be utilized as part of a complex bioassay for applications as diverse as proteomics and DNA sample preparation [2]. The DMFB control signals, termed actuation sequences (ASs), are computer-generated through a high-level synthesis flow [3]. A specification for the bioassay to be executed on-chip is written in a high-level descriptive language and passed to the synthesis software for processing. The synthesis flow typically consists of scheduling, placement, and routing phases, although alternate flows have been proposed which tackle all phases simultaneously [9], [10]. Common optimizations include reliability, testability, and execution time [3], [11]. The output of the synthesis flow is the AS: a time-indexed series of pin activations that can be applied directly to the DMFB.

Attackers can take advantage of the simple nature of DMFB ASs to great effect. The malicious modification of these control signals is called an *actuation tampering attack*, and were first reported in [4]. The term refers to the mechanism by which an attacker can achieve various malicious outcomes. Actuation tampering can be carried out through many different attack vectors, including alteration of data in program memory, modification of the software used to generate ASs, or physical injection of hardware faults. Such attacks are possible because ASs can be easily reverse-engineered to reveal the underlying protocol [12] and then modified to perform arbitrary fluid operations.

The earliest studies on actuation tampering considered general-purpose DMFBs requiring one IO pin from the driver circuitry for each electrode on-chip, a scheme termed *direct addressing*. However, one of the largest contributors to a DMFB's cost and complexity is the number of pins required to drive it [13]. Pin counts can quickly become impractical, even for modestly sized designs. For instance, a demonstrated

Manuscript received May 29, 2018; revised September 12, 2018; accepted October 30, 2018. Date of publication November 30, 2018; date of current version December 23, 2019. This work was supported in part by the Army Research Office under Grant W911NF-17-1-0320, in part by the National Science Foundation under Grant CNS-1833624, in part by NYU Center for Cyber Security (cyber.nyu.edu), and in part by NYU-AD Center for Cyber Security (sites.nyuad.nyu.edu/ccs-ad/). A preliminary version of this paper was published in Proc. IEEE/ACM Des. Autom. Conf., San Francisco, CA, USA, June 2018 [1]. This paper was recommended by Associate Editor S. Reda. *(Corresponding author: Jack Tang.)*

J. Tang and R. Karri are with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: jtang@nyu.edu; rkarri@nyu.edu).

M. Ibrahim and K. Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: mohamed.s.ibrahim@duke.edu; krish@duke.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2018.2883901

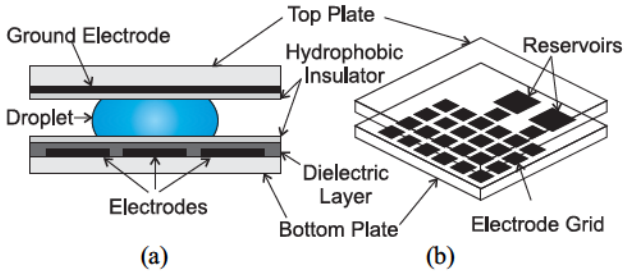

Fig. 1. Structure of a DMFB. (a) Side view. Droplet contact angle is modulated by an electric field applied to the electrodes. (b) General-purpose DMFB consists of a grid of electrodes upon which droplets are manipulated.

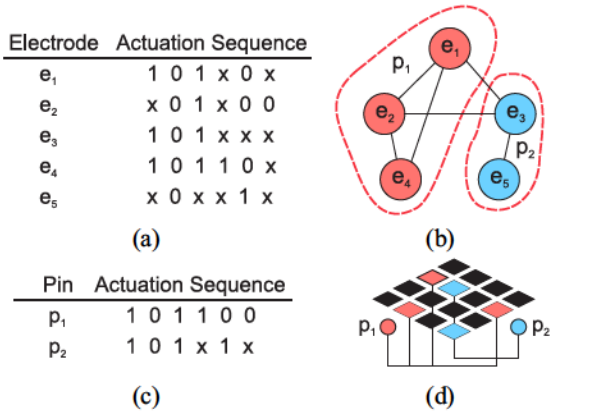

Fig. 2. Broadcast addressing. (a) Example AS for five electrodes. (b) Compatibility graph, with results of the clique selection outlined in dashed lines. (c) Resulting broadcast-addressed, pin-mapped AS. (d) Electrodes are physically wired together and brought out to pins for connection to a controller.

immunoassay DMFB targeted for point-of-care testing requires over 1000 pins [14]—which easily exhausts the number of pins on common MCU packages—if direct addressing is used. *Pin-constrained* DMFBs, on the other hand, reduce the pin count with a restriction on the droplet degrees of freedom. Pin-constrained DMFBs can be generated as the final step in the high-level synthesis flow (called *pin mapping*), or can considered in the overall biochip design [15]–[17]. The same immunoassay biochip described in [14], when pin-constrained, uses only 64 pins to drive over 1000 electrodes. Pin mapping reduces the degrees of freedom available for droplet manipulation, which also reduces the types of attacks that an attacker can execute. Under a pin-constrained DMFB, an attack may cause inadvertent droplet movements, which are detectable. Therefore, pin-mapped DMFBs are in some sense tamper-resistant.

This paper explores the concept of pin-mapping-as-tamper-resistance in detail through the following contributions.

1) We present the first security analysis of broadcast addressed, pin-constrained DMFB ASs and define the tamper resistance property with related definitions and security metrics.
2) We develop a new tamper-resistant pin mapper based on a greedy heuristic graph clustering algorithm.
3) We then develop a method to boost tamper resistance on sparse assays by inserting indicator droplets. An optimal integer linear programming (ILP) formulation and an ILP-based approximation algorithm are provided.
4) We present experimental evidence on several benchmark DMFB assays to show how the proposed methods improve security with modest overhead, as compared against prior pin mapping algorithms.

The rest of this paper is organized as follows. Section II presents a security analysis of pin-constrained DMFBs based on broadcast addressing. Security metrics are derived in Section III. Section IV develops a new pin mapping algorithm to optimize for tamper resistance. Section V describes ILP-based methodologies for boosting tamper resistance even further through insertion of indicator droplets. Experimental results are presented in Section VI and discussed in Section VII. We conclude in Section VIII.

II. Security Analysis

Here, we describe the concept of tamper resistance and how it arises as a result of pin-mapped DMFBs. We then derive the threat model based on reasonable assumptions about what makes an attack evident.

A. Preliminaries: Broadcast Addressing

Many post-synthesis pin mappers are based upon broadcast electrode-addressing schemes, which hardwires electrodes into pins receiving the same sequence of control signals [18]. Broadcast addressing is based on the concepts of *don't-care* values and *compatible sequences*.

1) Don't-Cares: On a DMFB grid, the movement of a single droplet requires a single pin activation (represented in the AS as a 1) to change the contact angle and initiate movement, and the deactivation (represented as 0) of the surrounding pins to ensure that the droplet does not inadvertently split. Any other electrode not directly involved in this transfer is a don't-care (represented by x), and can be held either high or low. The convention is chosen by the biochip designer, though typically it is held low.

2) Compatible Sequences: Two electrode ASs are compatible with each other if each value is either identical or at least one electrode contains a don't-care. Two compatible sequences can be combined into one by replacing don't-cares with the other electrode's activation value. This way, the two electrodes can be tied to the same pin receiving the same set of instructions. Hence, the term broadcast addressing.

Generation of a broadcast addressing scheme relies on graph-based representations of electrode relationships [18]. Vertices represent electrodes while edges represent relationships between compatible electrodes. Graph cliques can then be identified and partitioned, with the partition representing a collection of pins that can be shorted to a single driving pin (Fig. 2). This problem is NP-hard, but can be solved using heuristics [18]. Extensions to this basic concept include: reliability enhancement by reducing switching frequency and consequently reducing the degradation in contact angle [19], insertion of "ground vectors" for preventing residual charge [20], power consumption reduction through elimination of redundant actuation units (RAUs) [21]. We refer to these as toggle-aware, ground-vector-aware (GV-aware), and RAU-aware pin mappers, respectively, for consistency

with [22]. An optimal broadcast addressing scheme was developed in [23], achieving information-theoretical minimal pin counts. However, this scheme relies on the integration of digital logic in the biochip, which is impractical and yet to be demonstrated. Therefore, we consider such DMFBs to be outside the scope of those studied in this paper.

B. Redundant Units

We introduce the following concepts which we will use to analyze and design tamper-resistant pin-constrained DMFBs.

1) Redundant Units: Redundant Units (RUs) are don't-cares masked by either a 0 or a 1 due to the broadcast addressing scheme. This is a generalization of the RAU—first introduced in the power-aware pin mapper [21]—which is a don't-care that is masked by a 1. We also define a redundant deactuation unit (RDU) as a don't-care masked by a 0. For convenience, we define the function RU(p, t) which returns the number of RUs associated with pin p at timestep t, as well RAU(p, t) and RDU(p, t), which return number of RAU's and RDU's, respectively. These functions map pins and time-steps to integers that are pin-dependent, i.e., RU/RAU/RDU: $\mathcal{P} \times \mathcal{T} \rightarrow \mathbb{Z}$, where $\mathcal{P}$ is the set of all pins, $\mathcal{T}$ is the set of all assay time-steps. We also define ***RU***, ***RAU***, ***RDU***, as the set of all RU's, RAU's, and RDU's, respecitvely. A don't-care can be masked by either a 0 or a 1, but not both, so ***RAU*** and ***RDU*** are disjoint. The union of these two sets is equal to the set ***RU***. Therefore,

$$\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \text{RU}(p,t) = \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} (\text{RAU}(p,t) + \text{RDU}(p,t)). \quad (1)$$

2) Compatibility Degree: Compatibility degree (CD) measures how desirable it is to merge two electrode ASs and generalizes the binary concept of compatibility. Consider two electrode ASs AS_1 and AS_2, with m and n don't-cares, respectively. There are three ways in which the don't-cares can be arranged in time. In case 1 [Fig. 3(a)], all don't-cares overlap exactly and gives an attacker an opportunity for actuation tampering. In case 2 [Fig. 3(b)], some subset of the don't-cares overlap, and in case 3 [Fig. 3(c)], none overlap. Case 3 is the best-case scenario in terms of tamper resistance; there are no exposed don't-cares, and to target a masked don't-care, an attacker will risk modifying the normal execution of the assay. Therefore, we compute CD(e_1, e_2) as the number of RUs that result from the merging of two electrode ASs associated with electrodes e_1 and e_2.

C. Tamper Resistance

We claim that the tamper resistance of a DMFB is determined by the number and distribution of don't-cares in the pin-mapped AS. To illustrate, consider the attack in Fig. 4. This 15×19 DMFB is executing the InVitro 4 × 4 multiplexed diagnostic assay, which measures glucose (GLU), lactate (LAC), pyruvate (PYR), and glutamate (GLT) in plasma (PLA), serum (SER), saliva (SAL), and urine (URI) [3]. The attacker routes an extra plasma droplet along the red dashed line in order to alter the concentration of the sample being detected in DET3.

Normally, in a direct-addressed DMFB, the attack in Fig. 4 is trivial to implement and would have no consequences. In a pin-mapped design, all the electrodes in yellow are unintentionally activated in the course of routing the malicious droplet. This could result in the violation of design rules, such as in MIX29, where the droplet being mixed would be unintentionally split. Subsequently, the mix operation could fail, stopping the assay progression. Or, if a randomized checkpoint system is implemented, it may detect a stray droplet where none should exist [24]. An attacker can avoid this problem by targeting pins that drive a large number of don't-cares. Therefore, qualitatively, a tamper-resistant DMFB design will

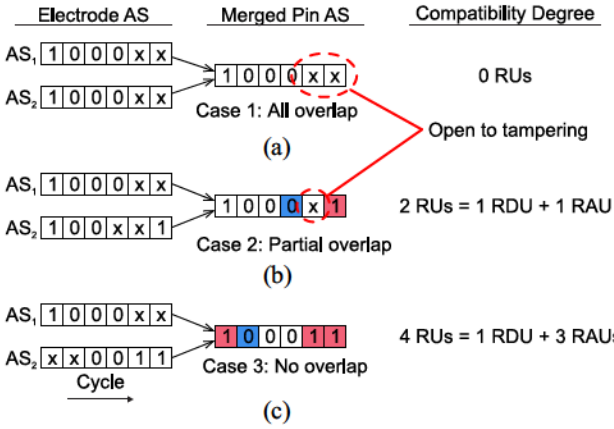

Fig. 3. AS merging. (a) Case 1: don't-cares exactly overlap. This is undesirable as an attacker can use the resulting pin-level don't-cares to insert malicious actuations. (b) Case 2: partial overlap. An RDU results when a broadcast 0 drives a don't-care, while a RAU is a 1 driving a don't-care. (c) Case 3: no overlap. This is the most desirable from a security standpoint; an attacker is forced to affect the operation of the assay since there are no don't-cares.

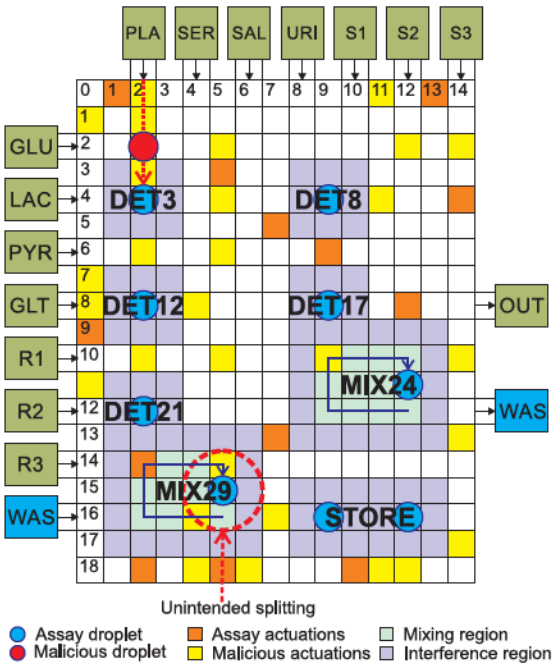

Fig. 4. Tamper resistance due to pin mapping: An attacker can modify the AS to produce the malicious droplet route in dotted lines. This attack would alter the final concentration reading. The yellow electrodes are unintentionally activated in the course of the attack due to pin mapping. Unintentional splitting may occur as a result of this, making the attack evident. This pin mapping was derived from the original broadcast clique-based strategy [18].

distribute don't-cares so attackers are constrained in their ability to make arbitrary droplet manipulations.

The practical interpretation of tamper resistance is that an attacker has fewer options available on a DMFB to make stealthy changes. Security is not a binary state (i.e., we do not seek a tamper-*proof* DMFB). Therefore, the metrics that we will define (Section III) will be normalized. If we consider an attacker interested in DoS, where the goal is to cause any disruption, then metrics defined in terms of maximums and minimums would be more appropriate. The stealthy attack model motivates this paper since this can lead to more disastrous outcomes, such as in a medical diagnostic application. DoS is often easily detectable and may not lead to any serious consequences.

D. Threat Model

We assume that the *attacker* is a remote party who can access the DMFB platform through the network. Controllers for DMFBs typically incorporate a network interface either by default (e.g., when using off-the-shelf embedded computers), or by design for firmware updates and sensor data processing. The attacker is able to conduct stealthy actuation tampering attacks, i.e., extract the synthesized ASs from memory, reverse-engineer them, and alter them. *The attacker does not want to be detected.* The extent of the alteration can range from simple augmentation or deletion of sequences, or can be as comprehensive as total replacement. Potential malicious actors and their motivations are discussed in [4]. The *defender* is the DMFB platform designer who wishes to ensure that any modifications to the ASs are easily detectable.

E. Attack Constraints

While the threat model grants an attacker tremendous capabilities, in practice, several factors will cause attacks to become evident to the end user. Therefore, arbitrary AS modifications may not be feasible due to the following constraints.

1) *Completion Time:* Assays may have completion times that are known to the end user. Relatively simple assays, e.g., sample preparation, can be assumed to execute in constant time. Such assays are commonly used as benchmarks in the DMFB literature and are studied in this paper. More complex assays that have multiple branching points depending on intermediate results may have variable execution times [25]. Still, an end user may suspect incorrect execution if an assay completes orders of magnitude faster or slower than their experience suggests is normal.
2) *Error Recovery:* DMFBs are known to be prone to several hardware faults. Cyberphysical integration has been proposed to detect and recover from errors [26]. The design of these mechanisms require fine tuning on the error tolerance, which may be exploitable for carrying out an attack. Furthermore, since placement of error recovery inspection points (i.e., *checkpoints*) is deterministic, a resourceful attacker could simply avoid making changes directly in critical paths [27].
3) *Intrusion Detection:* Intrusion detectors can monitor parts of the biochip execution that are not actively sensed by error recovery systems. Completely deterministic detection can in theory provide 100% security, but in practice, a low overhead scheme (e.g., randomized checkpoints [24]) must be implemented.
4) *Attack Surface:* We consider network-based attacks, where the AS can be recovered and modified at-will. Physical fault injection attacks are possible on DMFBs, but these typically present poor localization and would be unlikely to result in a stealthy attack.
5) *Reverse Engineering:* Many state-of-the-art designs for DMFBs store the AS in a format that has a one-to-one mapping between encoded bits and the biochip. Reverse engineering is thus straightforward [12]. If some mechanism were introduced to obfuscate the mapping, the attacker would not be able to make controlled changes to the assay.
6) *Unintentional Fouling:* An attacker who wishes to carry out stealthy attacks will want to minimize any unintended consequences. A droplet being routed across a biochip may contaminate electrodes in its wake [28]. As these contaminations accumulate, reactions may fail or concentrations may not meet threshold criteria during checkpoints. A failure would then be detected, thwarting the attack.

F. Threat Model Refinement

We refine the threat model due to the attack constraints.

1) *Increasing or Decreasing the Number of Time-Steps in the AS Is Prohibited:* This is to satisfy Constraint 1 for the nonconditional assays studied in this paper. Even slight variations in the AS length can result in noticeable execution time differences, as DMFB actuation periods are often on the order of milliseconds (which is coarse enough to be detected by a stopwatch). Therefore, the attack can only consist of *modifications* of the AS.
2) *The Number of Modifications to the AS Must be Minimized:* This is to avoid detection by either the end user, or detection by a checkpoint system (Constraints 2 and 3). In some cases, the effect of making an incremental change in the AS can be quantified; if a randomized checkpoint system is implemented, each additional change exponentially increases the probability of being detected [24].
3) *Modifications to the AS Will Preferentially Target Don't-Cares:* To do otherwise would be to modify activations (1s) inserted to control droplets or deactivations (0s) inserted as part of an interference region. On pin-constrained designs, modifying a pin-level control signal will change several electrode states. Therefore, if an attacker's goal is to control a single electrode, attacking a pin may cause unintentional changes to other electrodes, potentially causing a detectable change in assay execution.

III. Security Metrics

We now define several security metrics to mathematically capture the notion of tamper resistance. The first, coverage, will be useful for optimization. The second, pin disturbance, is more illustrative for interpreting the results of the optimization.

The third is based on randomization and provides the most direct indicator of how tamper resistance manifests itself against an attack.

A. Coverage

In Section II, it was stated that the number and distribution of don't-cares in the pin-mapped AS determines tamper resistance. One interpretation of this is that one should mask as many don't-cares as possible.

1) Redundant Unit Coverage: It is defined as the proportion of electrode-level don't-cares that are masked by pin-level actuations (i.e., RUs) over all pins and all assay time-steps. Therefore, it can be calculated as

$$\text{RUC} = \frac{\#\text{ redundant units}}{\#\text{ total don't-cares}}. \tag{2}$$

Redundant unit coverage (RUC) should be maximized. In the ideal case, coverage is equal to 100%, meaning that there are absolutely no exposed don't-cares for an attacker to leverage.

2) Proximity Coverage Class: A variation of RUC such that only electrodes within the vicinity of a droplet are counted

$$\text{PCC} = \frac{\#\text{ redundant units near any droplet}}{\#\text{ total don't-cares near any droplet}}. \tag{3}$$

Here, "near any droplet" means adjacent to the interference region along the x- or y-axis of a droplet. This coverage metric narrows the scope to attacks targeting functional droplets. That is, we exclude electrodes far from any functional droplets since they are unlikely to be used for manipulation attacks.

One subtle point about coverage metrics is that they are normalized. A pin mapper that generates a significant number of don't-cares can still have favorable coverage if those don't-cares are masked. That is, if we compare two pin mappers based only on the raw number of don't-cares, the one with fewer don't-cares may appear more secure. However, based on our qualitative definition of tamper resistance, this is not the case. Tamper resistance encompasses not only the fact that an attacker will try to leverage don't-cares, but also that they may attempt to tamper with functional droplets. Therefore, generating a large number of RUs can make up for the introduction of more don't-cares.

B. Pin Disturbance

We propose another set of metrics that more intuitively articulates the tamper resistance property. If we put ourselves in the perspective of an attacker, we are attempting to cause stealthy, directed changes on the biochip by altering the pin-level actuations. That is, while we want to control droplets precisely at the electrode level, our granularity of control is dictated at the pin level. Therefore, we introduce the following two metrics to measure the effect of pin-level changes.

1) Functional Pin Disturbance: The average number of electrodes that one would expect to disturb if a pin-level activation or deactivation is changed. A pin-level activation (1) is used to drive one or more droplets, and will also mask some underlying don't-care electrodes. If we deactivate a pin ($1 \rightarrow 0$), then all droplets being driven by the pin will no longer be bound to the electrode. That is, the droplet will float around the biochip until it is either absorbed by another droplet or finds an empty activated electrode. This disturbance is detectable. Conversely, if we change a deactivation to an activation ($0 \rightarrow 1$), the underlying electrodes in the vicinity of a droplet will pull on the functional droplets, causing them to distort and/or split. It is desirable to maximize this quantity. Let $\text{AU}(p, t)$ be a function that returns the number of electrode-level actuations and $\text{DU}(p, t)$ be a function that returns the number of electrode-level deactivations located at pin p and time-step t

$$\text{FPD} = \frac{1}{|\mathcal{P}||\mathcal{T}|} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} (\text{AU}(p, t) + \text{DU}(p, t)). \tag{4}$$

We can sum $\text{AU}(p, t)$ and $\text{DU}(p, t)$ since they are mutually exclusive on a pin-level basis.

2) Don't-Care Pin Disturbance: The average number of electrodes that can be controlled if one changes a pin-level don't-care into an activation ($x \rightarrow 1$). Don't-cares can be freely manipulated by an attacker. A pin-level don't-care that drives multiple electrodes gives an attacker more choices for electrodes to manipulate. Let $\text{DNT}(p, t)$ return the number of electrode-level don't-cares at pin p and time t that are driven by a pin-level don't-care. It is desirable to minimize this quantity, as this gives the attacker fewer options.

$$\text{DPD} = \frac{1}{|\mathcal{P}||\mathcal{T}|} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \text{DNT}(p, t). \tag{5}$$

C. Probability of Detection

The previous metrics, while well-founded, may seem abstruse. A more intuitive sense of the security afforded by tamper resistance can be obtained through performing a Monte Carlo experiment: we randomly generate attacks and execute them on-chip. Some proportion of these attacks will cause detectable interference with the assay, while others will not. The ratio of unsuccessful attacks against total attacks is the *probability of detection*, or $\Pr(D)$. The randomness of the generated attack is both spatial and temporal. We assume a malicious droplet will appear at any time-step and at any location on the biochip, and that it performs a random walk of random duration (subject to some maximum). At the end of the attack, we assume the droplet can then disappear. This is nonphysical, but simplifies the simulation and underestimates the probability of detection; more effort is required to route droplets to and from the random attack. The attack is detected if it conflicts with the AS.

IV. Tamper-Resistant Pin Mapping

We propose to increase tamper resistance through pin mapping. By selectively grouping electrodes together, isolated changes to the AS will become more difficult.

A. Problem Statement

The formal problem statement is described as follows.

Input: A DMFB architecture $\mathcal{A}$ consisting of a set of electrodes $\mathcal{B}$ and a set of electrode actuation sequences $\mathcal{AS}$.

Output: A pin-constrained DMFB design assigning each electrode $b \in \mathcal{B}$ to a set of pins $\mathcal{P}$, where $|\mathcal{P}| < |\mathcal{B}|$, and a set of pin-mapped actuation sequences $\mathcal{AS}_{\text{PM}}$.

Objective: Maximize the tamper resistance by maximizing the RUC.

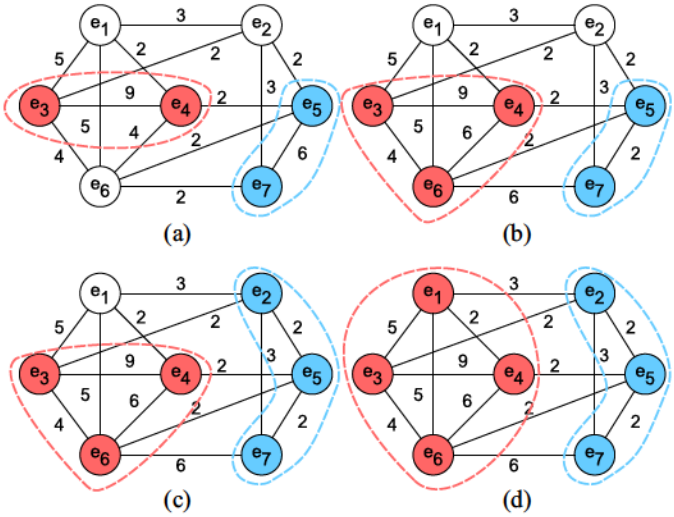

Fig. 5. Heuristic tamper-resistant pin mapping. (a) Initial phase of the heuristic algorithm. Vertices represent electrodes, weighted edges represent the CD between compatible electrodes. The highest weighted edges are selected for the initial set. (b) First iteration expands the red pin by selecting a clique-compatible vertex with highest compatibility. (c) Second iteration expands the blue pin. (d) Completion of the procedure.

B. Proposed Solution

The problem of grouping electrodes into pins can be modeled as a graph partitioning problem [18]. Here, we are also concerned with grouping electrodes into pins but now have imposed an additional constraint due to the desire to maximize tamper resistance. Based on our definition of CD, merging of highly compatible electrodes results in a tamper-resistant design. Therefore, the DMFB design is modeled by a graph $G = (V, E)$, where each vertex $v \in V$ represents an electrode on the DMFB array, and the set of edges E represent relationships between two compatible electrodes, similar to the original broadcast strategy. However, we now include an edge weighting function $w : E \rightarrow \mathbb{Z}$ that evaluates the CD between the two electrode ASs, and a "color" function $c : V \rightarrow \mathbb{Z}$, which represents the pin assignment.

By grouping electrodes that are highly compatible, we promote solutions that increase the number of RUs, and therefore result in a more tamper-resistant design. The grouping of high dimensional data represented by graphs is known as the graph clustering problem [29]. In particular, our specific problem of forming k number of pins out of the graph vertices associated with a distance function (i.e., "the compatibility degree") is known as the minimum k-clustering problem, which has been shown to be NP-hard [30]. We therefore propose a greedy heuristic graph-based algorithm to solve the tamper resistance optimization problem (Fig. 5), which proceeds as follows.

1) Initialize the graph and weights (G, w).
2) Form an initial guess p_{init} for the number of pins to form.
3) Starting from the highest-weighted edges, assign their corresponding two vertices (electrodes) to a pin by setting their color to a unique value, and skipping vertices that have already been assigned.
4) Repeat forming new pins until p_{init} pins have been formed.

Input: Set of electrodes $\mathcal{B}$, actuation sequence $\mathcal{AS}$, initial pin count p_{init}
Output: Map $c(b)$ from electrodes to pin

$(G = (V, E), w) \leftarrow$ initGraph($\mathcal{AS}$)
$c(v) \leftarrow \varnothing, \forall v \in V$
$sortedEdges \leftarrow$sortDescending(E, w)
for $i \in \{0, 1, 2, ..., p_{init}\}$ **do**
  $v =$getVertices($sortedEdges$.pop())
  if $c(v) = \varnothing$ **then**
    $c(v) = i$
  end if
end for
while $\exists c(v) = \varnothing$ or no colors assigned in last iteration **do**
  for $i \in \{0, 1, 2, ..., p_{init}\}$ **do**
    for each $\{v \in V : c(v) = i\}$ **do**
      $neighbors \leftarrow$getNeighbors(v)
      $candidates \leftarrow \varnothing$
      for n in $neighbors$ **do**
        if $c(n) = \varnothing$ **then**
          $candidates$.add(compatibilityDegree(n, v))
        end if
      end for
      $c(\max(candidates)) \leftarrow i$
    end for
  end for
end while
$currentPin \leftarrow p_{init} + 1$
while $\exists c(v) = \varnothing$ **do**
  for each $\{v \in V : c(v) = \varnothing\}$ **do**
    $c(v) = currentPin$
    $neighbors \leftarrow$getNeighbors(v)
    $candidates \leftarrow \varnothing$
    for n in $neighbors$ **do**
      if $c(n) = \varnothing$ **then**
        $candidates$.add(compatibilityDegree(n, v))
      end if
    end for
    $c(\max(candidates)) \leftarrow currentPin$
    $currentPin+ = 1$
  end for
end while
return $c(b)$

Fig. 6. Tamper-resistant pin mapper pseudocode.

5) Expand the pins by greedy iteration. For each pin, examine all neighbor vertices that are not-yet assigned, and clique-compatible with the pin, add the vertex with highest compatibility, then move on to the next pin.
6) Repeat until either all electrodes are assigned or no more valid neighbors can be added.
7) Repeat the overall procedure on the remaining unassigned electrodes, and failing that, assign each an individual pin.

This is a fast heuristic algorithm that attempts to group together pins that are most compatible, with complexity $\mathcal{O}(|V|)$ since an electrode is greedily assigned at each step. The initial guess for the pin count can be established through trial-and-error, or using knowledge of typical broadcast-addressed pin counts. The procedure is summarized in Fig. 6.

V. Boosting Tamper Resistance Using Indicator Droplets

Intuitively, if an AS is sparse, then we would expect that the tamper-resistant pin mapper would be unable to produce a high coverage rate. This is due to a lack of actuations and deactuations to use for masking. The sparseness of an AS depends on how the bioassay has been scheduled and placed (i.e., how many concurrent operations are running), and whether routing

compaction is implemented (i.e., concurrently routing droplets for the next operation phase [31]). Even if the scheduler and placer attempted to maximize concurrency, the bioassay may be constrained by critical nodes. For example, a final mix reaction may require that all previous reactions are completed first. The resulting AS would be sparse during the final cycles.

We propose to transform a bioassay such that tamper resistance is increased through the insertion of indicator droplets. If the indicator droplets are inserted properly, tampering with the AS will cause them to deviate from their expected paths. Error recovery sensors can easily detect these deviations.

It is important to note that introducing excess droplets to the biochip will degrade reliability. Droplets may accidentally pick up residues and distribute contaminants across the biochip (*cross-contamination* [32]). Excessive actuations will be introduced [33]. However, we can justify this if we are able to quantify the effect of the excess droplets and set it as a tuning parameter. This will be useful to the system designer as the amount of excess usage that is tolerable will increase as surface coatings are improved. Furthermore, if the indicator droplets are wash droplets, they will have the added benefit of cleaning the electrodes in their wake.

A. Problem Statement

We state the tamper resistance boosting problem as follows.

1) *Input:* A synthesized actuation sequence $\mathcal{AS}$.
2) *Output:* A modified actuation sequence $\mathcal{AS}_{\text{BST}}$.
3) *Objective:* Increase tamper resistance through insertion of indicator droplets while minimizing the number of excess actuations.
4) *Constraints:* Indicator droplets should obey all design rules and not interfere with functional droplet operation.

B. ILP-Based Indicator Droplet Insertion

We propose an ILP formulation to exactly solve the indicator droplet insertion problem. An electrode's state is determined by the voltage that is actually applied (the electrical state, $\mathcal{E}$), and its usage (the functional state, $\mathcal{F}$). We represent this combined state $\mathcal{S}$ with the notation $(\mathcal{E}/\mathcal{F})$ and code them as follows.

1) *State 0 (0/0):* Electrode is electrically held low (0) so as to enforce an interference region, or is deleted but part of an interference region.
2) *State 1 (1/1):* Electrode is electrically held high (1) so as to keep a droplet in place.
3) *State 2 (x/x):* Electrode can be driven either high or low and is functionally a don't-care.
4) *State 3 (1/x):* Electrode is being driven high through pin mapping, but is functionally a don't-care, i.e., an RAU.
5) *State 4 (0/x):* Electrode is being driven low through pin mapping, but is functionally a don't-care, i.e., an RDU.
6) *State 5 (z/x):* Electrode is disconnected and not part of an interference region.

This notation is summarized in Table I. Note that other combinations, such as $(x/0)$ and $(1/z)$ do not arise in practice.

We now define some conventions and sets to aid the ILP formulation. The range of time-steps is $\mathcal{T} = \{0, 1, 2, \ldots, T_{\max}\}$

TABLE I
ILP MODEL ELECTRODE STATE NOTATION

State $\mathcal{S}$	Electrical/Functional State $(\mathcal{E}/\mathcal{F})$	Description
0	$(0/0)$	Enforcement of IR region
1	$(1/1)$	Enforcement of droplet
2	(x/x)	Unmasked don't-care
3	$(1/x)$	Redundant activation unit
4	$(0/x)$	Redundant deactivation unit
5	(z/x)	Disconnected, not part of an IR region

while the possible electrode states are $\mathcal{S} = \{0, 1, 2, 3, 4, 5\}$ according to Table I. The coordinate convention is chosen with the origin in the top-left corner. We also add a padded area around the biochip to facilitate formulation of the ILP constraints; without doing so, we would have to specify special constraints for each of the eight edge cases: four for each corner, and four for each side. For example, the coordinate of the top-left corner on a biochip is $(0, 0)$, but becomes $(1, 1)$ when it is padded. For an $m \times n$ biochip, the range of coordinate values $\mathcal{C} = \{0, 1, 2, \ldots, n+1\} \times \{0, 1, 2, \ldots, m+1\}$. After solving the model, we simply discard the border region. To simplify notation, when we sum over a single variable, we will assume it means to sum over all possible values. We also assume that the AS can be interpreted as a function $\mathcal{AS}(t, x, y) \in \{0, 1, 2\}^{\mathcal{T} \times m \times n}$, where 0 is a deactivation, 1 is an activation, and 2 is a don't-care.

We now define our multiobjective optimization problem. The problem is to set the state of the electrodes in such a way as to minimize both the number of excess actuations and the number of unmasked don't-cares. We assume that an initial bootstrap phase is permitted, where an arbitrary number of indicator droplets can be routed onto the chip prior to the start of the assay. Introduce a binary variable $\delta_{t,x,y,s}$ which takes value 1 if electrode at coordinate (x, y) at time-step t has state s. The objective is

$$\min : \alpha \sum_{t} \sum_{(x,y)} \delta_{t,x,y,2} + \beta \sum_{t} \sum_{(x,y)} \delta_{t,x,y,1} \tag{6}$$

where the first term is the sum of all nonmasked don't-care terms, and the second term the sum of all electrode activations. This optimizes the tamper resistance and the reduction in reliability, respectively. α and β are weighting factors to be set by the system designer. The optimization is subject to the following constraints.

1) *Actuation Sequence:* We fix electrodes associated with a functional droplet and its IR region.

$$\delta_{t,x,y,1} = 1, \forall t \in \mathcal{T}, (x, y) \in \mathcal{C} : \mathcal{AS}(t, x, y) = 1 \tag{7}$$
$$\delta_{t,x,y,0} = 0, \forall t \in \mathcal{T}, (x, y) \in \mathcal{C} : \mathcal{AS}(t, x, y) = 0. \tag{8}$$

2) *State Exclusivity:* Each electrode at each time-step can only occupy a single state.

$$\sum_{s} \delta_{t,x,y,s} = 1, \quad \forall t \in \mathcal{T}, (x, y) \in \mathcal{C}. \tag{9}$$

3) *Pin Mapping:* Enforcement of pin mapping requires two related constraints. The first is defined at the pin level. The sum of all $(0/x)$ states in a pin cannot be equal

TABLE II
SYMBOLS USED IN ILP MODEL

Symbol	Elements	Description
$\mathcal{T}$	$\{0, 1, 2, ..., T_{max}\}$	Set of all time-steps
$\mathcal{X}$	$\{0, 1, 2, ..., n+1\}$	Set of all padded x-coordinates
$\mathcal{Y}$	$\{0, 1, 2, ..., m+1\}$	Set of all padded y-coordinates
$\mathcal{C}$	$\mathcal{X} \times \mathcal{Y}$	Set of all padded biochip coordinates
$\mathcal{P}$	$\{p_0, p_1, p_2, ..., p_{max}\}$	Set of all pins
p_i	$\{(x_0, y_0), (x_1, y_1), .., (x_{\lvert p_i \rvert}, y_{\lvert p_i \rvert})\}$	Set of coordinates of electrodes in pin i
$\mathcal{Z}(p_i)$	$\{(x_0, y_0, x_1, y_1), (x_1, y_1, x_2, y_2), ...\}$	Function returning set of all transitive pairs of electrode coordinates in p_i
$\mathcal{R}$	$\{(x_0, y_0), (x_1, y_1), .., (x_{\lvert \mathcal{R} \rvert}, y_{\lvert \mathcal{R} \rvert})\}$	Set of coordinates of deleted electrodes
$\mathcal{W}(t, x, y)$	$\{(x_0, y_0), ..., (x_7, y_7)\}$	Function returning set of surrounding neighbor coordinates
$\mathcal{N}(x, y)$	$\{(x_0, y_0), ..., (x_3, y_3)\}$	Function returning set of N/S/E/W neighbors
$\mathcal{I}$	$\{(x_0, y_0), (x_1, y_1), .., (x_{\lvert \mathcal{I} \rvert}, y_{\lvert \mathcal{I} \rvert})\}$	Coordinates of all IO ports

to the number of electrodes in the pin; to do so would imply that there is no driving electrode.

$$\sum_{(x,y) \in p_i} \delta_{t,x,y,3} < |p_i|, \quad \forall t \in \mathcal{T}, \forall p_n \in \mathcal{P}. \tag{10}$$

Similarly, the sum of all $(1/x)$ states in a pin cannot equal the number of electrodes in the pin.

$$\sum_{(x,y) \in p_i} \delta_{t,x,y,4} < |p_i|, \quad \forall t \in \mathcal{T}, \forall p_n \in \mathcal{P}. \tag{11}$$

The second is defined at the electrode level. An electrode can be electrically high by being in state 1 OR 4. We can model this as the sum of state 1 and 4 for a particular electrode. For any two electrodes in a pin, if one electrode is being electrically driven by a 1, then the other electrode must be driven by a 1. We model this through equality of the two electrode states.

$$\begin{aligned} &\delta_{t,x1,y1,1} + \delta_{t,x1,y1,4} \\ &\quad = \delta_{t,x2,y2,1} + \delta_{t,x2,y,2,4} \\ &\qquad \forall t \in T, \forall (x_1, y_1, x_2, y_2) \in \mathcal{Z}(p_i), \forall p_i \in \mathcal{P} \end{aligned} \tag{12}$$

where $\mathcal{Z}(p_i)$ returns all transitive pairs of electrode coordinates in pin p_i. That is, if electrode A is driven by a logic 1 forces electrode B to a logic 1, and electrode B forces electrode C to logic 1, then electrode A will also force electrode C to logic 1. This constraint is unnecessary for pins with only a single electrode.

4) *Removed Electrodes:* Some electrodes are not used in the course of the protocol's execution. The pin mapper will "delete" these electrodes by not connecting any pins to them. The system designer can then choose not to fabricate these electrodes. For modeling purposes, we assume that these electrodes still exist, but allow them to only be in state 0 or 5 by fixing all other state variables to 0. Let $\mathcal{R}$ denote the set of coordinates corresponding to electrodes removed during pin mapping.

$$\sum_t \delta_{t,x,y,s} = 0, \forall k \in \{1, 2, 3, 4\}, \forall (x, y) \in \mathcal{R}. \tag{13}$$

For all other electrodes, we can disallow setting them to the disconnected state (5).

$$\sum_t \delta_{t,x,y,5} = 0, \quad \forall (x, y) \in \mathcal{C} \setminus \mathcal{R}. \tag{14}$$

5) *Interference Regions:* If an electrode has an indicator droplet (i.e., has state 1), it must keep an intereference region of 0 deactivations around it. Functional assay droplets are derived from synthesis and already have appropriate IR enforcement. They also must be able to merge and split. Applying IR constraints to functional droplets will cause infeasibility. To avoid this, let $\mathcal{W}(t, x, y)$ be a function returning the set of coordinates surrounding an electrode at (t, x, y). We define the following for all don't-cares in the AS.

$$\begin{aligned} 8 \cdot \delta_{t',x',y',1} \leq &\sum_{(x,y) \in \mathcal{W}(t',x',y')} \delta_{t',x,y,0}, \\ &\forall t' \in \mathcal{T}, (x', y') \in \mathcal{C} : \mathcal{AS}(t', x', y') = 2. \end{aligned} \tag{15}$$

6) *Continuity:* Droplets must maintain their presence between time-steps. Let $\mathcal{N}(x, y)$ return the set with coordinates (x, y) and its direct North/South/East/West neighbors.

$$\begin{aligned} \delta_{t',x',y',1} \leq &\sum_{(x,y) \in \mathcal{N}(x',y')} \delta_{t+1,x,y,1} \\ &\forall t' \in \mathcal{T}, (x', y') \in \mathcal{C} : \mathcal{AS}(t, x, y) = 2. \end{aligned} \tag{16}$$

7) *Dispense/Waste Ports:* Indicator droplets can only be introduced or consumed at dispense and waste ports, respectively. Activations for all other electrodes must be disallowed. Let $\mathcal{I}$ be the set of coordinates for all dispense and waste ports. The constraint is not imposed at time-step 0 to generate an initial bootstrap solution.

$$\delta_{t,x,y,1} = 0, 1 \leq t \leq T_{\max}, \forall (x, y) \in \mathcal{C} \setminus \mathcal{I}. \tag{17}$$

C. Iterative ILP-Based Sliding Window Approximation

The exact ILP-based formulation (Table II) is optimal, but is clearly intractable for all but the simplest DMFBs. This mainly stems from the time complexity of the model used; all time-steps are considered in forming the optimal solution, which can be in the tens of thousands. To overcome this, we propose several approximations and an ILP-based sliding window algorithm, which is summarized in Fig. 7.

We approximate the minimization of unmasked don't-cares by introducing the concept of *impact*, which is measured for each electrode and time-step of an AS. An electrode

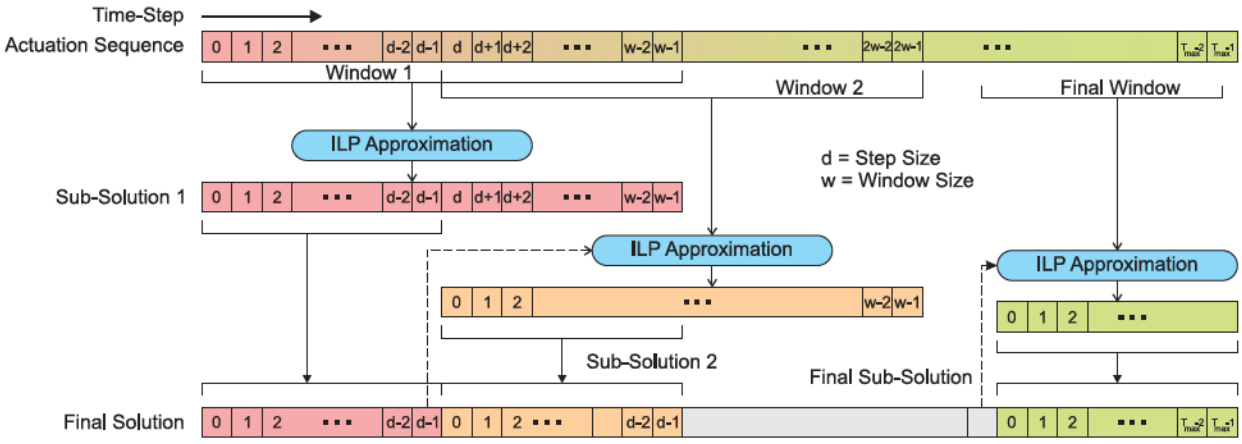

Fig. 7. ILP-based sliding window approximation algorithm. We subdivide the problem into smaller manageable chunks, and solve them using the simplified ILP approximation algorithm. We construct the final solution by truncating and appending the subsolutions. The final window may be smaller than the other windows if the AS length is not an integer multiple of the window size.

Input: Actuation sequence $\mathcal{AS}$
Output: Impact vector $\boldsymbol{w}_{t,x,y}$
1: $\boldsymbol{w}_{t,x,y} \leftarrow \varnothing, \forall v \in \mathcal{T}, (x,y) \in \mathcal{C}$
2: **for** $\forall t \in \mathcal{T}, (x,y) \in \mathcal{C} \setminus \mathcal{R}$ **do**
3: testActuation $\leftarrow$insertDroplet($\mathcal{AS}, t, x, y$)
4: $\boldsymbol{w}_{t,x,y} \leftarrow$ countRUs(testActuation)
5: **end for**
6: **return** $\boldsymbol{w}_{t,x,y}$

Fig. 8. Pseudocode for calculating electrode impact.

Input: Actuation sequence $\mathcal{AS}(t,x,y)$
Output: Map $\mathcal{V}(t,x,y)$ indicating validity of electrode (x,y) at time-step t
1: $\mathcal{V}(t,x,y) \leftarrow \varnothing$
2: **for** $\forall t \in \mathcal{T}$ **do**
3: **for** each pin-connected electrode $\forall (x,y) \in \mathcal{C} \setminus \mathcal{R}$ **do**
4: **if** $\mathcal{AS}(t,x,y)$ is don't-care or $(1/x)$ and all neighbors in $\mathcal{W}(t,x,y)$ equals don't-care or $(0/x)$ **then**
5: $\mathcal{V}(t,x,y) \leftarrow 1$
6: **end if**
7: **end for**
8: **end for**
9: **return** $\mathcal{V}(t,x,y)$

Fig. 9. Pseudocode for valid electrode preprocessing.

has high impact if, when we place a droplet on it, a large number of don't-cares are masked. Define a helper function insertDroplet($\mathcal{AS}, t, x, y$) which returns a state vector $s \in \mathcal{S}^{(m+1)\times(n+1)}$ representing the biochip at time t if we insert a droplet at position (x, y). Also define a function countRUs(s) which counts the number of RUs in a state vector. Then we simply proceed to insert test droplets and count RUs for each electrode and each time-step, collecting the results into a vector $\boldsymbol{w}_{t,x,y}$. Therefore, the procedure takes $\mathcal{O}(\mathcal{XYT})$. The calculation of impact is summarized in Fig. 8.

The simplified ILP model depends on preprocessing the AS to find valid locations for placing an indicator droplet. Let $\mathcal{V}(t,x,y)$ be a function taking on value 1 if electrode (x, y) at time-step t is a don't-care or masked activation $(1/x)$ with sufficient interference region clearance, and 0 otherwise. The procedure for preprocessing has worst-case complexity $\mathcal{O}(\mathcal{XYT})$ and is summarized in Fig. 9.

We then derive our approximate ILP model as follows. Define the binary variable $\delta_{t,x,y}$ to take value 1 if we are to add an indicator droplet at coordinate (x, y) and time-step t, 0 otherwise. The objective is to maximize impact by placing a single indicator droplet.

$$\max : \sum_{t} \sum_{(x,y)} \boldsymbol{w}_{t,x,y} \delta_{t,x,y}. \tag{18}$$

Subject to the following constraints.

1) *Valid Electrodes:* Due to the presence of functional droplets and pin mapping, only some electrodes are eligible to add droplets. We fix invalid locations to 0.

$$\delta_{t,x,y} = 0, \quad \forall t \in \mathcal{T}, (x,y) \in \mathcal{C} : \mathcal{V}(t,x,y) = 1. \tag{19}$$

2) *Single Droplet Routing:* We can only add a single droplet in each time-step. This is a self-imposed constraint so as to speed up computation. Additional droplets are added during the iteration phase.

$$\sum_{(x,y)} \delta_{t,x,y} \leq 1, \quad \forall t \in \mathcal{T}. \tag{20}$$

3) *Continuity:* This is defined identically as the original ILP formulation in (16).

Despite our simplifications, a large number of time-steps will still lead to intractable models. To overcome this, we can use the ILP approximation as the basis of an iterative sliding window algorithm (Fig. 7). Define a window width of w time-steps and a step size of d time-steps, and iterate by solving the ILP approximation for the first w time-steps. We take the first d solutions, and then solve the next window of time-steps offset by d. We repeat and keep adding solutions until the overall problem is solved. Each iteration of this algorithm yields a boosted AS with one additional indicator droplet. We can iteratively repeat the procedure on the boosted AS until no more can be added (Fig. 11).

VI. Experimental Results

We evaluated our tamper-resistant pin mapper against the broadcast addressing [18], RAU-aware [21], GV-aware [20], and toggle-aware [19] pin mappers using four benchmark assays: 1) PCR; 2) InVitro 4x4; 3) Protein; and 4) Protein Split 5. The benchmark simulation data was generated with

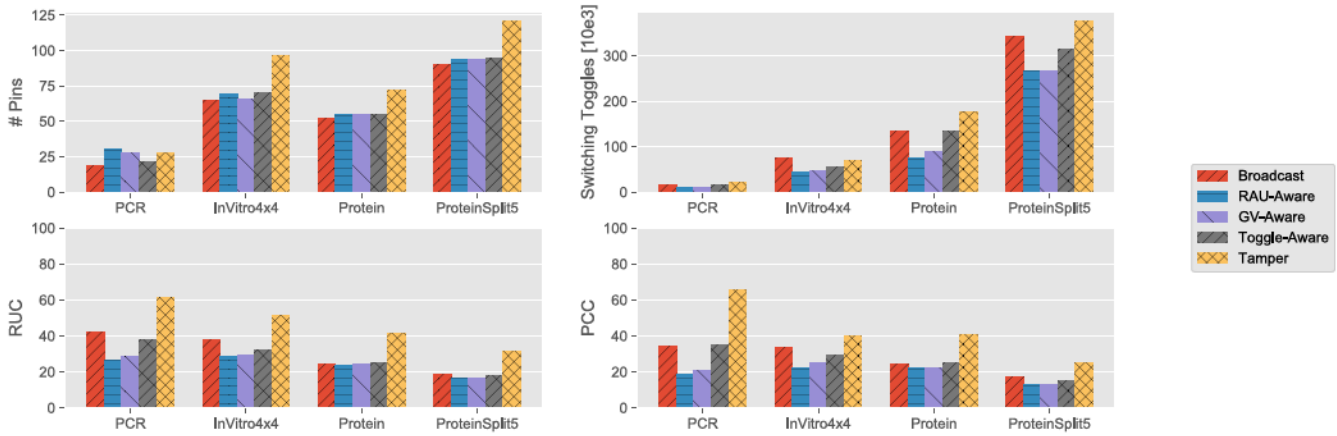

Fig. 10. Baseline comparison of the tamper-resistant pin mapper against Broadcast [18], RAU-Aware [21], GV-Aware [20], and Toggle-Aware [19] pin mappers. The number of pins is increased, but still within an acceptable range for integration on a PCB. Switching toggles (SW) are increased. This paper achieves, on average, 39.6% higher RUC than the next-best prior work, the Broadcast clique pin mapper.

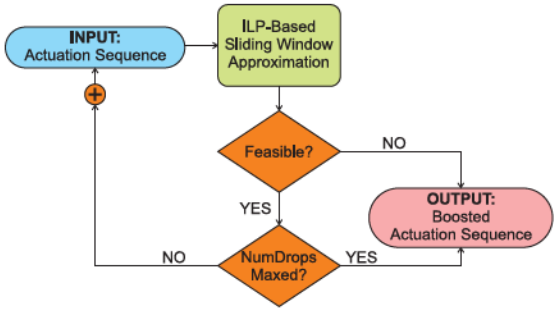

Fig. 11. Iterative ILP-based sliding window approximation algorithm. Indicators are successively added to the AS until either the model is infeasible or the maximum number of indicators have been reached.

the open-source MFStaticSim tool [22] using a 15 × 19 DMFB array, virtual topology placer and Roy maze router. We imported this data for analysis and implemented our tamper-resistant pin mapper in MATLAB. We then implemented the proposed ILP-based sliding window approximation algorithm in Python 3.6.4 using Gurobi 7.5 as our ILP solver. We used an Intel Core i7-8700 3.2 GHz machine with 16GB RAM. The window size was set to 100 time-steps and the step size was set to 50 time-steps. The most complex assay, ProteinSplit5, took more than one day to complete. However, after the initial indicator droplet was placed, subsequent droplets took only a few hours to route.

A. Baseline Performance

We summarize baseline performance in Fig. 10, which shows pin counts, switching toggles (SW) measured in thousands as an inversely related indicator of power and reliability, RUC and proximity coverage class (PCC) measured in percentages. We see that tamper-resistant pin mapper achieves, on average, 39.6% higher RUC than the next-best prior work, which is the broadcast clique pin mapper. When we restrict attacks to the PCC, coverage can be as high as 73.9% for some assays using the tamper-resistant pin mapper. Most of the pin mappers achieve coverage rates of less than 50% across all assays. At the same time, this paper's pin counts are modestly increased while switching activity is increased but remains on the same order. We note that the pin counts are still reasonable for implementation on a PCB. Additionally, compatibility in the RAU-aware pin mapper was enabled between ASs with opposite polarities, with the reasoning that an inverter could be inserted in the biochip. While this does save on pins, most commercially available DMFBs and prototypes today feature completely passive biochips. Therefore, the pin-count savings may not be realizable in practice.Therefore, we conclude that the proposed algorithm achieves its goal and is able to produce a quantifiably more tamper-resistant pin-constrained DMFB.

B. Performance With Indicator Droplets

Performance for the boosted assays are shown in Fig. 12. As expected, inserting indicator droplets incrementally increases the security performance. Each pin mapper exhibits a similar rate of change. Since the tamper-resistant pin mapper starts out with better metrics, it continues to beat the other pin mappers as indicator droplets are added. In our experimental data, pin mappers for the PCR assay typically deleted around 200 out of 285 electrodes, meaning that there is very little degree of freedom for droplet routing. This is exhibited in the saturation of the curves after two or three indicator droplets; this shows that the solver fails to find ways to insert indicator droplets. The ability to plot the change in SW with indicator droplets allows one to select an appropriate number of indicator droplets for a given piece of hardware.

C. Probability of Detection

Probability of detection is shown in the bottom row of Fig. 12. We performed 1000 Monte Carlo experiments for each assay and pin mapper combination, setting the maximum random walk length to 10. Increasing the random walk only disadvantages the attacker, as this gives more time-steps for a detection system to discover the malicious droplet [24], [34]. We observe that the tamper-resistant pin mapper achieves better baseline probability of detection than all other pin mappers. As indicator droplets are added, the performance gap decreases, although the general trend is maintained. This demonstrates that arbitrary attacks can be detected with high

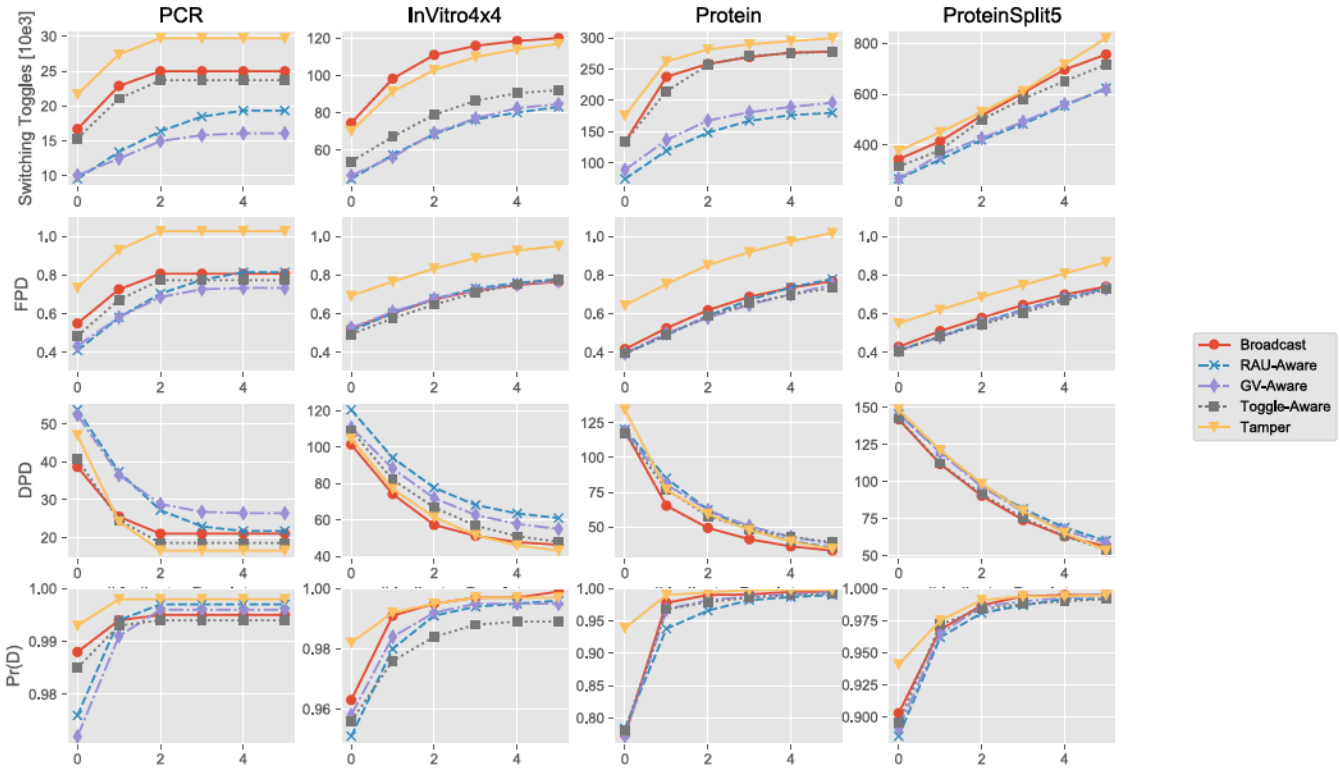

Fig. 12. Security metrics versus number of added indicator droplets. Successively adding indicator droplets improves the security metrics but also increases excess SW. In some pin mappers, the PCR assay saturates after a few indicator droplets are added. Probability of detection Pr(D) can be made to be nearly 1 in many cases.

probability, approaching 1 in many cases, and that the tamper-resistant pin mapper is advantageous if we require that no indicator droplets are to be added. The only exception to the trend is in the InVitro4x4 assay, where after the fifth indicator droplet is added, the Broadcast addressing scheme actually outperforms the tamper-resistant pin mapper.

VII. Discussion

The basis of tamper resistance is the idea that certain actions by an attacker are detectable. It is conceivable that a clever adversary could design the attack in such a way that it meets all the correctness criteria set by execution checkpoints. In this case, it is recommended that a randomized checkpoint system be utilized [24], [34]. Randomized checkpoints give a probability of evasion that is parameterized by attack length L. We denote this quantity with hat notation as $\Pr(\hat{E})$. We may quantify the interaction of randomized checkpoints with tamper resistance by converting the probability of detection metric into a probability of evasion as $\Pr(E) = 1 - \Pr(D)$. The overall system's probability of evasion, denoted by boldface as $\Pr(\boldsymbol{E})$, is the probability that the attacker evades both the randomized checkpoints and evades a collision with the unaltered AS. These are independent events, therefore we have

$$\Pr(\boldsymbol{E}) = \Pr\left(\hat{E}\right)(1 - \Pr(D)). \tag{21}$$

Of course, we may convert this into a probability of detection by finding the complement as $\Pr(\boldsymbol{D}) = 1 - \Pr(\boldsymbol{E})$. The probability of detection encountered in practice can be quite high. On the PCR assay, if we assume the worst-case probability of evasion from a randomized checkpoint system that monitors 10% of the electrodes, we have $\Pr(\hat{E}) = 0.90$ [24]. Tamper-resistant pin mapping with no indicator droplets results in $\Pr(D) = 0.993$, which gives $\Pr(\boldsymbol{D}) = 0.9937$.

A. Comparison With Other Countermeasures

A number of existing techniques can be adopted to make tampering infeasible, but with some practical shortcomings.

1) *Encryption* is often the first defense that comes to mind for both protecting the privacy and integrity of an AS. However, encryption in hardware systems is not foolproof. Insecure systems may arise as a result of poor key management schemes [35]. At some point, the AS *must* be decrypted to be executed on hardware. Attackers can then tamper with the AS after it has been decrypted. Encryption only protects the AS when the device is at rest. Moving the encryption to hardware bears significant implementation costs [36]. Furthermore, encryption implementations may be susceptible to side-channel attacks [37]. Proper use of encryption is nontrivial to achieve.
2) *Cryptographic hashes* can provide a unique signature for an AS pattern, but will fail if the signature is just as susceptible to attack as the AS—which is often the case. This situation bears similarity to software download hashes, which are a good idea for checking against random transmission errors, but poorly suited when an adversarial threat model is adopted.
3) *Code signing* is a commonly employed method for ensuring the integrity of firmware. In fact, the main reason for Stuxnet's success was the lack of code

signing [38]. However, this simple countermeasure can fail; major corporations have been known to lose control of their private signing keys [39]–[41].

In contrast, incorporating security measures in the pin mapping phase of the DMFB design flow is highly advantageous as no extra circuitry or control hardware is required. Our approach has modest pin count overhead, which is acceptable since the PCB layer count can form a substantial portion of overall system cost anyway [13]. Furthermore, this is a hardware-based technique that is not susceptible to attacks that take advantage of network-enabled controllers. And by inserting indicator droplets, we substantially increase tamper resistance using an essentially free resource: water.

B. Related Prior Work

Research on security and trustworthiness of microfluidic systems is still in its early stages. Subtle result manipulation attacks on glucose assays were described in [4], while DMFB supply chain security was evaluated in [42]. Randomized checkpoint systems were proposed to detect attacks in real time [24]. In the realm of intellectual property protection, reverse-engineering attacks were systematized in the BioChipWork framework [12]. An IP protection scheme based on the concept of a "fluidic multiplexer" was used to realize "fluidic encryption," which requires the application of a fluid-based secret key for the assay to function properly [43]. A more practical method to lock bioassays through insertion of dummy mix-split operations was recently proposed [44]. A PUF-based digital rights management scheme was proposed in [45], while a method to localize attacks on ASs was proposed in [46]. Beyond digital microfluidics, transposer-based routing fabrics have been studied for their security properties under fault injection attacks [47].

VIII. Conclusion

We presented the first study of DMFB pin mappers as a tamper resistance mechanism. The restriction on droplet movements imposed by pin mappers simultaneously lowers an attacker's ability to arbitrarily route droplets, and causes undesirable side-effects on other droplets existing on the chip. We introduced the redundant unit coverage security metric to describe the masking of don't-cares. Experimental results show that existing pin mapping algorithms, while optimizing for reliability and power consumption, lead to poor tamper resistance. A new pin mapping algorithm was proposed to increase masking effects, and ILP-based techniques were developed to insert indicator droplets to boost tamper resistance even further with modest, quantifiable pin count and switching overhead.

Tamper-resistant pin mapping is a simple design-time technique to harden a DMFB design against actuation tampering attacks. It is hardware-based and is therefore intrinsic to the operation of the DMFB, mitigating the effects of an attack should high-level security mechanisms fail. It comes at no cost to the designer, as a pin mapping algorithm must be chosen regardless of security concerns. There is, however, a fundamental tradeoff in control complexity and switching toggles required for a given level of security. This tradeoff is quantifiable and thus tunable by the system designer.

The proposed methodology ensures high coverage rates as a post-synthesis processing step. While optimizing for tamper resistance during the high-level synthesis phases could in theory provide better performance, we believe it is more productive to leverage all of the rich literature on DMFB synthesis. Furthermore, methodologies that do not subdivide the synthesis tasks often have long runtimes despite being based on heuristics [48].

This paper does have some limitations that can be addressed as future work. We have considered pin mappers only as a post-synthesis processing step. Therefore, the optimization problem is highly constrained. This explains why 100% coverage is not achievable. It is conceivable that even better tamper resistance properties could be achieved if other aspects of the synthesis flow, such as placement and routing, were optimized for tamper resistance. The experimental results show that some overhead is required to optimize for tamper resistance. This can be adjusted through trial and error, but it would be preferable to fine-tune the balance as a multiobjective optimization problem.

We note that the assays studied in this paper are static and do not feature conditional execution [49]. The concept of tamper resistance through pin mapping can be adapted to conditional assays, but would require some care in its implementation. The conditionally executed ASs must be synthesized in such a way that it is compatible with a given pin mapping. The original broadcast addressing scheme provides one such method for adding additional operations to an existing pin-mapped design through appropriate scheduling [18]. Additionally, the security metrics would need to be redefined since a reference AS is not available as a reference for correctness. It should be possible to quantify the "effort" required to move a droplet from a random source to a random destination.

Other future work includes investigating methods to deter DoS attacks, and to extend the work to be applicable to emerging micro-electrode-dot-array biochips [50], [51].

References

[1] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Tamper-resistant pin-constrained digital microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Conf.*, San Francisco, CA, USA, 2018, pp. 1–6.

[2] K. Choi, A. H. Ng, R. Fobel, and A. R. Wheeler, "Digital microfluidics," *Annu. Rev. Anal. Chem.*, vol. 5, no. 1, pp. 413–440, 2012.

[3] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 4, p. 1, 2008.

[4] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security assessment of cyberphysical digital microfluidic biochips," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 3, pp. 445–458, May/Jun. 2016.

[5] Baebies, Inc. (2017). *Baebies Seeker*. [Online]. Available: http://baebies.com/products/seeker/

[6] The Wall Street Journal. (Mar. 2016). *Theranos Results Could Throw Off Medical Decisions, Study Finds*. [Online]. Available: http://www.wsj.com/articles/theranos-results-could-throw-off-medical-decisions-study-finds-1459196177

[7] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab Chip*, vol. 2, no. 2, pp. 96–101, 2002.

[8] S. K. Cho, H. Moon, and C.-J. Kim, "Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits," *J. Microelectromech. Syst.*, vol. 12, no. 1, pp. 70–80, Feb. 2003.

[9] F. Su and K. Chakrabarty, "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Conf.*, 2005, pp. 825–830.

[10] T. Xu and K. Chakrabarty, "Integrated droplet routing and defect tolerance in the synthesis of digital microfluidic biochips," *ACM. J. Emerg. Technol. Comput. Syst.*, vol. 4, no. 3, p. 11, 2008.
[11] K. Chakrabarty, "Design automation and test solutions for digital microfluidic biochips," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 1, pp. 4–17, Jan. 2010.
[12] H. Chen, S. Potluri, and F. Koushanfar, "Biochipwork: Reverse engineering of microfluidic biochips," in *Proc. IEEE Int. Conf. Comput. Design*, Boston, MA, USA, Nov. 2017, pp. 9–16.
[13] D. T. Grissom, J. McDaniel, and P. Brisk, "A low-cost field-programmable pin-constrained digital microfluidic biochip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 11, pp. 1657–1670, Nov. 2014.
[14] R. Sista *et al.*, "Development of a digital microfluidic platform for point of care testing," *Lab Chip*, vol. 8, no. 12, pp. 2091–2104, 2008.
[15] Y. Luo and K. Chakrabarty, "Design of pin-constrained general-purpose digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 9, pp. 1307–1320, Sep. 2013.
[16] D. Grissom and P. Brisk, "A field-programmable pin-constrained digital microfluidic biochip," in *Proc. IEEE/ACM Design Autom. Conf.*, 2013, p. 46.
[17] C. C.-Y. Lin and Y.-W. Chang, "ILP-based pin-count aware design methodology for microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 9, pp. 1315–1327, Sep. 2010.
[18] Y. Zhao, T. Xu, and K. Chakrabarty, "Broadcast electrode-addressing and scheduling methods for pin-constrained digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 7, pp. 986–999, Jul. 2011.
[19] S.-T. Yu, S.-H. Yeh, and T.-Y. Ho, "Reliability-driven chip-level design for high-frequency digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 4, pp. 529–539, Apr. 2015.
[20] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2011, pp. 448–455.
[21] T.-W. Huang, H.-Y. Su, and T.-Y. Ho, "Progressive network-flow based power-aware broadcast addressing for pin-constrained digital microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Conf.*, 2011, pp. 741–746.
[22] D. Grissom *et al.*, "An open-source compiler and PCB synthesis tool for digital microfluidic biochips," *Integr. VLSI J.*, vol. 51, pp. 169–193, Sep. 2015.
[23] T. A. Dinh, S. Yamashita, and T.-Y. Ho, "An optimal pin-count design with logic optimization for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 4, pp. 629–641, Apr. 2015.
[24] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Secure randomized checkpointing for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 6, pp. 1119–1132, Jun. 2018.
[25] M. Ibrahim, K. Chakrabarty, and K. Scott, "Synthesis of cyberphysical digital-microfluidic biochips for real-time quantitative analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 5, pp. 733–746, May 2017.
[26] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 59–72, Jan. 2013.
[27] Y. Zhao, T. Xu, and K. Chakrabarty, "Integrated control-path design and error recovery in the synthesis of digital microfluidic lab-on-chip," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 6, no. 3, p. 11, 2010.
[28] H. Yao, Q. Wang, Y. Shen, T.-Y. Ho, and Y. Cai, "Integrated functional and washing routing optimization for cross-contamination removal in digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1283–1296, Aug. 2016.
[29] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007.
[30] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theor. Comput. Sci.*, vol. 38, pp. 293–306, 1985.
[31] T.-W. Huang and T.-Y. Ho, "A fast routability- and performance-driven droplet routing algorithm for digital microfluidic biochips," in *Proc. IEEE Int. Conf. Comput. Design*, 2009, pp. 445–450.
[32] Y. Zhao and K. Chakrabarty, "Cross-contamination avoidance for droplet routing in digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 6, pp. 817–830, Jun. 2012.
[33] J. K. Park, S. J. Lee, and K. H. Kang, "Fast and reliable droplet transport on single-plate electrowetting on dielectrics using nonfloating switching method," *Biomicrofluidics*, vol. 4, no. 2, 2010, Art. no. 024102.
[34] J. Tang, M. Ibrahim, and K. Chakrabarty, "Randomized checkpoints: A practical defense for cyberphysical microfluidic systems," *IEEE Des. Test.*, to be published, doi: 10.1109/MDAT.2018.2863118.
[35] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, *Recommendation for Key Management Part 1: General (Revision 4)*, document SP 800 57, NIST, Gaithersburg, MD, USA, pp. 1–147, 2012.
[36] N. Sklavos, "On the hardware implementation cost of crypto-processors architectures," *Inf. Security J. Glob. Perspective*, vol. 19, no. 2, pp. 53–60, 2010.
[37] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2002, pp. 29–45.
[38] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.
[39] T. Mendelsohn. (2016). *Secure Boot Snafu: Microsoft Leaks Backdoor Key, Firmware Flung Wide Open*. [Online]. Available: https://arstechnica.com/information-technology/2016/08/microsoft-secure-boot-firmware-snafu-leaks-golden-key/
[40] D. Goodin. (2015). *In Blunder Threatening Windows Users, D-Link Publishes Code-Signing Key*. [Online]. Available: https://arstechnica.com/information-technology/2015/09/in-blunder-threatening-windows-users-d-link-publishes-code-signing-key/
[41] B. Schneier. (2012). *Sony Playstation 3 Master Key Leaked*. [Online]. Available: https://www.schneier.com/blog/archives/2012/10/sony_playstatio.html
[42] S. S. Ali, M. Ibrahim, J. Rajendran, O. Sinanoglu, and K. Chakrabarty, "Supply-chain security of digital microfluidic biochips," *Computer*, vol. 49, no. 8, pp. 36–43, Aug. 2016.
[43] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Microfluidic encryption of on-chip biochemical assays," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Shanghai, China, Oct. 2016, pp. 152–155.
[44] S. Bhattacharjee, J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Locking of biochemical assays for digital microfluidic biochips," in *Proc. IEEE Euro. Test Symp.*, Bremen, Germany, 2018, pp. 1–6.
[45] C.-W. Hsieh, Z. Li, and T.-Y. Ho, "Piracy prevention of digital microfluidic biochips," in *Proc. Asia South Pac. Design Autom. Conf.*, Chiba, Japan, Jan. 2017, pp. 512–517.
[46] P. Roy and A. Banerjee, "A new approach for root-causing attacks on digital microfluidic devices," in *Proc. IEEE Asian Hardw. Orient. Security Trust Symp.*, Yilan, Taiwan, Dec. 2016, pp. 1–6.
[47] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Security trade-offs in microfluidic routing fabrics," in *Proc. IEEE Int. Conf. Comput. Des.*, Boston, MA, USA, Nov. 2017, pp. 25–32.
[48] D. T. Grissom and P. Brisk, "Fast online synthesis of digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 3, pp. 356–369, Mar. 2014.
[49] D. Grissom, C. Curtis, and P. Brisk, "Interpreting assays with control flow on digital microfluidic biochips," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 3, p. 24, 2014.
[50] G. Wang *et al.*, "Field-programmable lab-on-a-chip based on micro-electrode dot array architecture," *IET Nanobiotechnol.*, vol. 8, no. 3, pp. 163–171, 2013.
[51] G. Wang, D. Teng, and S.-K. Fan, "Digital microfluidic operations on micro-electrode dot array architecture," *IET Nanobiotechnol.*, vol. 5, no. 4, pp. 152–160, Dec. 2011.

Jack Tang (M'18) received the B.S. degree in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2006, the M.S. degree in electrical engineering from San Jose State University, San Jose, CA, USA, in 2012, and the Ph.D. degree in electrical engineering from New York University, Brooklyn, NY, USA, in 2018.

His current research interests include cyberphysical microfluidic systems, MEMS, and analog circuit design and their application toward secure and trustworthy hardware.

Mohamed Ibrahim (M'18) received the B.Sc. and M.Sc. degrees (with Distinction and Hons.) in computer and systems engineering from Ain Shams University, Cairo, Egypt, in 2010 and 2013, respectively, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from Duke University, Durham, NC, USA, in 2017 and 2018, respectively.

As a Graduate Researcher, he researched on the design and optimization of trustworthy lab-on-chip (LoC) systems to support critical DNA-analysis flows. He was a Visiting Scholar with the Technical University of Munich, Munich, Germany, and the University of Bremen, Bremen, Germany. He spent a total of three years as a Research and Development Engineer in the semiconductor industry, where he researched on design-for-test and post-silicon validation methodologies for several system-on-chip (SoC) designs. He has published over 30 papers on these topics in premier journals and refereed conference proceedings. His current research interests include SoC design and embedded systems, electronic design automation of LoC systems, Internet-of-Bio-Things, security and trust of bio-systems, and machine-learning applications of bio-systems.

Dr. Ibrahim was a recipient of the 2018 Council of Graduate Schools/ProQuest Distinguished Dissertation Award in Mathematics, Physical Sciences, and Engineering, the Best Paper award at the 2017 IEEE/ACM Design, Automation, and Test in Europe Conference, the 2017 Post-Doctoral Mobility Award from the Technical University of Munich, Germany, two ACM Conference Travel Awards from ACM-SIGBED in 2016 and ACM-SIGDA in 2017, and Duke Graduate School Fellowship in 2013. He served as a Technical Program Committee Member for the 2019 IEEE/ACM Design, Automation, and Test in Europe Conference, and the 2019 IEEE International Conference on Very Large Scale Integration Design, and as an expert reviewer for many conferences and journals.

Krishnendu Chakrabarty (F'08) received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, Kharagpur, India, in 1990 and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1992 and 1995, respectively.

He is currently the William H. Younger Distinguished Professor and the Department Chair of electrical and computer engineering, and a Professor of computer science with Duke University, Durham, NC, USA. He is a Research Ambassador with the University of Bremen, Bremen, Germany, and a Hans Fischer Senior Fellow with the Institute for Advanced Study, Technical University of Munich, Munich, Germany. His current research interests include testing and design-for-testability of integrated circuits and systems, microfluidic biochips, hardware security, machine learning for fault diagnosis and failure prediction, and neuromorphic computing systems.

Prof. Chakrabarty was a recipient of the National Science Foundation CAREER Award, the Office of Naval Research Young Investigator Award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS Donald O. Pederson Best Paper Award in 2015, the *ACM Transactions on Design Automation of Electronic Systems* Best Paper Award in 2017, and over a dozen best paper awards at major conferences, the IEEE Computer Society Technical Achievement Award in 2015, the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award in 2017, the Semiconductor Research Corporation Technical Excellence Award in 2018, the IEEE Test Technology Technical Council Bob Madge Innovation Award in 2018, and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur, in 2014. He was the 2018 Japan Society for the Promotion of Science Fellowship in the "Short Term S: Nobel Prize Level" category. He has been a Distinguished Visitor of the IEEE Computer Society from 2005 to 2007 and from 2010 to 2012, a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2006 to 2007 and from 2012 to 2013, and an ACM Distinguished Speaker from 2008 to 2016. He served as the Editor-in-Chief of IEEE Design and Test of Computers from 2010 to 2012 and *ACM Journal on Emerging Technologies in Computing Systems* from 2010 to 2015. He is currently serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is also an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON MULTISCALE COMPUTING SYSTEMS, and *ACM Transactions on Design Automation of Electronic Systems*. He is a fellow of ACM, as fellow of AAAS, and a Golden Core Member of the IEEE Computer Society.

Ramesh Karri (SM'11) received the B.E. degree from Andhra University, Visakhapatnam, India, and the Ph.D. degree in computer science and engineering from the University of California at San Diego, San Diego, CA, USA.

He is currently a Professor of electrical and computer engineering with New York University (NYU), Brooklyn, NY, USA. He co-directs the NYU Center for Cyber Security and leads the Cyber Security thrust of the New York State Center for Advanced Telecommunications Technologies with NYU. He co-founded Trust-Hub. He co-founded the Embedded Systems Challenge, the global red-team–blue-team hardware hacking event. He has authored over 200 journal and conference publications, including tutorials on trustworthy hardware in the IEEE Computer Society and the *Proceedings of the IEEE*. He has delivered invited keynotes, talks, and tutorials on hardware security and trust at venues, such as IEEE/ACM Design Automation Conference (DAC), IEEE/ACM Design, Automation, and Test in Europe, IEEE Very Large Scale Integration Test Symposium (VTS), IEEE International Test Conference (ITC), the IEEE International Conference on Computer Design (ICCD), IEEE North Atlantic Test Workshop, IEEE Latin–American Test Workshop, and CROSSING. His current research interests include hardware cybersecurity, covering trustworthy ICs, processors and cyberphysical systems, security-aware computer-aided design, test, verification, validation and reliability, nanotechnology security, metrics, benchmarks, hardware cybersecurity competitions, and additive manufacturing security.

Prof. Karri was a recipient of the Humboldt Fellowship, the National Science Foundation CAREER Award, Awards at ITC 2014, the ACM Computer and Communications Security Conference in 2013, the IEEE Symposium on Defect and Fault-Tolerant Nano VLSI Systems (DFTS) in 2013, and the VLSI Design in 2012, the ACM Student Research Competitions at DAC 2012, the ICCAD 2013, and the DAC 2014, the ACM Grand Finals 2013, and the Kaspersky Challenge and Embedded Security Challenge, and nominated for best paper awards at ICCD 2015 and DFTS 2015 for hardware cybersecurity. He organizes the annual Embedded Security Challenge, a red-team/blue-team hardware security competition with NYU. He co-founded and served as the Chair of the IEEE Computer Society Technical Committee on Nanoscale Architectures (NANOARCH). He co-founded and serves on the Steering Committee of the IEEE/ACM Symposium on NANOARCH. He served as the Program Chair and the General Chair of several conferences, including ICCD, the IEEE Symposium on Hardware Oriented Security and Trust (HOST), the IEEE Symposium on Defect and Fault-Tolerant Nano VLSI Systems (DFTS), IEEE/ACM NANOARCH, RFIDsec, and ACM WiSec. He serves on several program committees, including DAC, ICCAD, HOST, ITC, VTS, IEEE European Test Symposium, ICCD, IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale Era. He is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the *ACM Journal of Emerging Technologies in Computing*, the *ACM Transactions on Design Automation of Electronic Systems*, the IEEE TRANSACTIONS ON EMERGING TECHNOLOGIES IN COMPUTING, the IEEE DESIGN AND TEST, and the IEEE EMBEDDED SYSTEMS LETTERS. He also served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2010 to 2014 and IEEE ACCESS from 2015-2017. He was an IEEE Computer Society Distinguished Visitor from 2013 to 2015. He was on the Executive Committee of the IEEE/ACM Design Automation Conference, initiating and leading the Security@DAC initiative from 2014 to 2017.